

LEVEL II

12

AD A108805



Accelerated Convergence of Structured Banded Systems Using Constrained Corrections

Karl Kneile
Sverdrup Technology, Inc.

December 1981

Final Report for Period October 1980 — August 1981

DTIC
ELECTE
DEC 22 1981
S **D**
E

Approved for public release; distribution unlimited.

ARNOLD ENGINEERING DEVELOPMENT CENTER
ARNOLD AIR FORCE STATION, TENNESSEE
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE

81 12 23 153

NOTICES

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Technical Information Center

References to named commercial products in this report are not to be considered in any sense as an endorsement of the product by the United States Air Force or the Government.

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

APPROVAL STATEMENT

This report has been reviewed and approved:

Keith L. Kushman

KEITH L. KUSHMAN
Directorate of Technology
Deputy for Operations

Approved for publication:

FOR THE COMMANDER

Marion L. Laster

MARION L. LASTER
Director of Technology
Deputy for Operations

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AEDC-TR-81-22	2. GOVT ACCESSION NO. AD-A108 805	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ACCELERATED CONVERGENCE OF STRUCTURED BANDED SYSTEMS USING CONSTRAINED CORRECTIONS		5. TYPE OF REPORT & PERIOD COVERED Final Report, October 1980 - August 1981
7. AUTHOR(s) Karl Kneile, Sverdrup Technology, Inc., AEDC Group		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Arnold Engineering Development Center/DOT Air Force Systems Command Arnold Air Force Station, TN 37389		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Arnold Engineering Development Center/DOS Air Force Systems Command Arnold Air Force Station, TN 37389		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 65807F
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1981
		13. NUMBER OF PAGES 20
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Available in Defense Technical Information Center (DTIC).		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) multigrid interations potential flow		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Descretization of fluid flow equations results in a structured banded system of equations. Iterative methods used to solve such a system transfer information rapidly between neighboring points but exhibit slow global communication when the system is large. This phenomenon results in a slow convergence with the asymptotic stages showing a spatially smooth behavior for the errors, iterative corrections, and residuals. This smooth behavior is exploited		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT, Concluded.

by approximating the error and corrections with an interpolation over a selected subset of the grid. A variational form is used to compress the original system of equations into a smaller system. This compression is repeated through several levels of grid sizes to obtain a dramatic improvement in convergence rate. A relaxation parameter is dynamically calculated at each step with a negligible increase in computational effort. Although the method was developed for solving full potential flow using a variational principle, it is applicable to other problems with the structured banded property.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC). The Air Force project manager was Dr. Keith Kushman, AEDC/DOT. The results of the research were obtained by Sverdrup Technology, Inc., AEDC Group, operating contractor for propulsion testing at the AEDC, AFSC, Arnold Air Force Station, Tennessee, under Project Number P32A-AH1. The manuscript was submitted for publication on September 14, 1981.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A	

CONTENTS

	<u>Page</u>
1.0 INTRODUCTION	5
2.0 METHOD OF CONSTRAINED CORRECTIONS	5
3.0 DYNAMIC RELAXATION	7
4.0 INTERPOLATED FORM	7
5.0 MULTIGRID	8
6.0 STRUCTURED BANDING	10
7.0 SMOOTHING PASS	13
8.0 TEST CASE	13
9.0 SUMMARY	18

ILLUSTRATIONS

Figure

1. One-Dimensional Interpolation	11
2. Two-Dimensional Interpolation	12
3. Two-Dimensional Interpolation in Split Form	12
4. Aspect Ratio versus Convergence Rates	17

TABLES

1. Convergence Rates for Basic Algorithm	14
2. Convergence Rates for Alternate Boundary Conditions	14
3. Convergence Rates for ADI Smoothing	15
4. Convergence Rates for Weighted Passes	15
5. Convergence Rates Without Dynamic Relaxer	16

NOMENCLATURE	19
--------------------	----

1.0 INTRODUCTION

The purpose of this report is to describe an efficient iterative method for solving a structured banded system of equations. Although the method was developed for a full potential flow program, it will be presented in general terms applicable to a wide range of problems. The central issue here is the solution of a large linear system of equations. The linear system may arise directly in the problem or may result from an iteration in a nonlinear problem. For large two-dimensional (2-D) and three-dimensional (3-D) applications, this linear system becomes increasingly expensive to solve directly. As a result, efficient iterative methods have become attractive for large problems. In the nonlinear cases, these iterations may be effectively merged to improve convergence rates.

Conventional iterative methods (Jacobi, Gauss-Seidel, ADI, etc.) rapidly reach a state where convergence rates are limited by the large eigenvalues of the system. This phenomenon is especially restrictive for large problems. Various approaches have been tried to accelerate convergence. Relaxation made some modest gains, but obtaining an optimum or near-optimum parameter was sometimes difficult. Others have tried more elaborate iterative methods [incomplete Crout, strongly implicit procedure (SIP), and SIP/conjugate gradient] with considerable success. However, the most dramatic improvements have been seen recently with the revival of multigrid concepts.

The method presented in this report uses a basic iteration step (incomplete Crout reduction), a dynamic relation step, and a multigrid concept of constraining iterative corrections.

2.0 METHOD OF CONSTRAINED CORRECTIONS

The method of constrained corrections uses a variational form of the problem. This variational form may be part of the problem definition or may be artificially created as described later.

The discretized variational form may be represented as $L(\phi)$, where L is a scalar function of the n component vector ϕ . The components of ϕ are obtained by solving the n simultaneous equations

$$F = \partial L / \partial \phi = 0 \quad (1)$$

An iterative procedure (Newton's) for solving this system is as follows:

$$A\delta = r \quad (2)$$

where

$$\delta = \phi_{i+1} - \phi_i,$$

$$r = -F_i,$$

and

$$A = \partial F / \partial \phi|_i = \partial^2 L / \partial \phi^2|_i$$

For linear problems the iteration process is trivial and ends with the first iteration.

If a variational principle is not part of the problem, one can define

$$L^*(\delta) = \frac{1}{2} \delta^T A \delta - \delta^T r \quad (3)$$

and use

$$\delta L^* / \delta \delta = 0 \quad (4)$$

as the variational form. This is equivalent, both here and in later considerations, to using

$$\partial L(\phi_i + \delta) / \partial \delta = 0 \quad (5)$$

coupled with Newton's method if (5) is nonlinear in δ . The method of constrained corrections defines δ as

$$\delta = Ck \quad (6)$$

The vector k has $p < n$ components. The matrix C prescribes each component of δ as a linear combination of the components of k . It will be assumed that C is of rank p (i.e., the columns of C are linearly independent). This is equivalent to imposing $(n - p)$ linear combinations of the components of δ as zero. That is,

$$C^* \delta = 0 \quad (7)$$

where C^* is an $(n - p) \times n$ matrix. It is more convenient to use these constraints in the form of Eq. (6).

Substitution of (6) into the variational form results in the system of equations

$$C^T A C k = C^T r \quad (8)$$

for the unknown k vector. The δ vector is then obtained from (6). With judicious selections of C and k , the convergence rate can be substantially improved.

3.0 DYNAMIC RELAXATION

Consider again the basic linear system, (2). When using an iterative method, one obtains an approximate δ denoted by δ_a . By letting C in (6) be the vector δ_a , the relaxation parameter, k , is then given by

$$k = \delta^T r / \delta^T A \delta_a \quad (9)$$

The iteration then takes the form

$$\phi_{i+1} = \phi_i + k \delta_a \quad (10)$$

The residual r in (9) is the original residual vector using ϕ_i and is not obtained from using $\phi_i + \delta_a$.

4.0 INTERPOLATED FORM

It is easier to describe this form for one-dimensional problems. The components of ϕ are associated with a positional value along this dimension. As mentioned earlier, conventional iterative methods rapidly reach an asymptotic convergence limited by the larger eigenvalues. It is well known that these iterations rapidly remove the smaller wavelength components, leaving the smooth, longer wavelength components. Conventional multigrid methods exploit this smoothness to justify using a coarse grid operator. This paper will also take advantage of this smoothing property, but will direct emphasis toward the smoothness of the correction vector, δ . A basis vector, δ_b , is selected. The correction vector is then constrained to the following form:

$$\delta = C \delta_b = \begin{bmatrix} I \\ -B \end{bmatrix} \delta_b \quad (11)$$

In actual practice the components of δ_b are interspersed within δ , and the rows of B are merged with the rows of the identity matrix. The above representation (separated I and B) will be used to simplify notation. For this special case the constraints take the form

$$C^* \delta = \begin{bmatrix} B^T \\ -I \end{bmatrix} \delta = 0 \quad (12)$$

The B matrix represents interpolation coefficients for the nonbasis components. Solutions of

$$C^T A C \delta_b = C^T r \quad (13)$$

coupled with (11) will then "solve the original problem" subject to the constraints. The effectiveness of this constrained form depends upon the form of interpolation used, the smoothness of δ , and the difficulty in solving the new system, (13). The smaller the dimension of δ_b , the simpler system (13) is to solve. However, more iterations are needed to precondition the smoothness required for effective interpolation.

The dynamic relaxation step described in the previous section can be used to improve overall convergence. The relaxation factor may be calculated using the basis δ_b , as follows:

$$k = \delta^T (C^T r) / \delta_b^T (C^T A C) \delta_b \quad (14)$$

5.0 MULTIGRID

The constrained corrections method described in the previous section can be easily adapted to a multigrid concept. A nested sequence of basis vectors is defined by

$$\begin{aligned} \delta_0 &= \delta \\ \delta_{i-1} &= C_i \delta_i \quad i = 1, 2, \dots, m \\ C_i &= \begin{bmatrix} I \\ -B_i \end{bmatrix} \end{aligned} \quad (15)$$

where δ_0 represents the original fine grid and δ_m the coarsest level with the fewest components. The B_i values are interpolation coefficients from the i^{th} level to the $(i-1)$ level. The above interpolations may be combined to form

$$\begin{aligned} \delta_0 &= D_i \delta_i \\ D_i &= C_1 C_2 \dots C_i = D_{i-1} C_i \end{aligned} \quad (16)$$

The constrained system of equations is

$$D_i^T A D_i \delta_i = D_i^T r \quad (17)$$

These equations easily lend themselves to the following iterative algorithm.

A smoothing pass is made on the fine grid system.

$$A \delta = r \quad (18)$$

The ϕ vector is updated by (10), whereby (18) now represents the next iterative pass. A compression step is taken to obtain the system

$$A_1 \delta_1 = r_1 \quad (19)$$

where

$$A_1 = D_1^T A D_1$$

and

$$r_1 = D_1^T r$$

A smoothing pass is now made on this system. The ϕ vector is again updated, and the next iterative pass is taken at the second level,

$$A_2 \delta_2 = r_2 \quad (20)$$

where

$$A_2 = D_2^T A D_2$$

and

$$r_2 = D_2^T r$$

The process is repeated down through the coarsest level.

The preceding describes a multigrid cycle. This cycle is repeated until sufficient convergence is obtained. Many variations of the above algorithm are possible. A few of these are compared in Section 8.0.

A computational advantage can be obtained from the nesting or recursive definitions of the D_i . The next level system can be calculated directly from the current level,

$$A_{i+1} = C_{i+1}^T A_i C_{i+1} \quad (21)$$

$$r_{i+1} = C_{i+1}^T r_i$$

It is not necessary to calculate the updated residuals at the fine grid level. They may be calculated at the current i^{th} level and then compressed down one level.

6.0 STRUCTURED BANDING

Consider those problems where a quantity ϕ is to be determined over a 2-D or 3-D space. The space is discretized by an $(n_1 \text{ by } n_2)$ or $(n_1 \text{ by } n_2 \text{ by } n_3)$ grid. The A matrix in (2) takes a structured banded form. That is, only a few of the diagonals of A have nonzero elements. The particular structure of A depends upon the approximations used in describing the original equations at the grid points. This report will cover the details for a nine-diagonal structure typical of a 2-D finite element approach. Adaptations to other type problems should not pose any difficulties. Occasional comments concerning other type problems will be made at appropriate places.

The structure of A can be considered as a block tridiagonal system where the blocks are also tridiagonal. This simple structure allows computationally efficient smoothing algorithms. It is therefore desirable to maintain this structure through the multigrid levels. This imposes limitations on the interpolation matrix B. For simplicity, a 1-D case will be described first. The structured banded A matrix is tridiagonal. In order to maintain this tridiagonal structure, the interpolation for any point must be limited to a nearest neighbor principle. That is, an interpolated point is a function of only the two neighboring points, one on each side. This suggests a linear interpolation.

Figure 1 depicts a case where the δ vector has been smoothed. The corresponding C matrix has the following form (note: The rows of B and I are merged):

$$C^T = \begin{bmatrix} 1 & 1/2 & & & & & & & \\ & 1/2 & 1 & 2/3 & 1/3 & & & & \\ & & & 1/3 & 2/3 & 1 & 3/4 & 1/2 & 1/4 \\ & & & & & & 1/4 & 1/2 & 3/4 & 1 \end{bmatrix} \quad (22)$$

In the typical multigrid patterns where every other point is used, C has the following form for a nine-point to five-point compression.

$$C^T = \begin{bmatrix} 1 & 1/2 & & & & & & & \\ & 1/2 & 1 & 1/2 & & & & & \\ & & & 1/2 & 1 & 1/2 & & & \\ & & & & & 1/2 & 1 & 1/2 & \\ & & & & & & 1/2 & 1 \end{bmatrix} \quad (23)$$

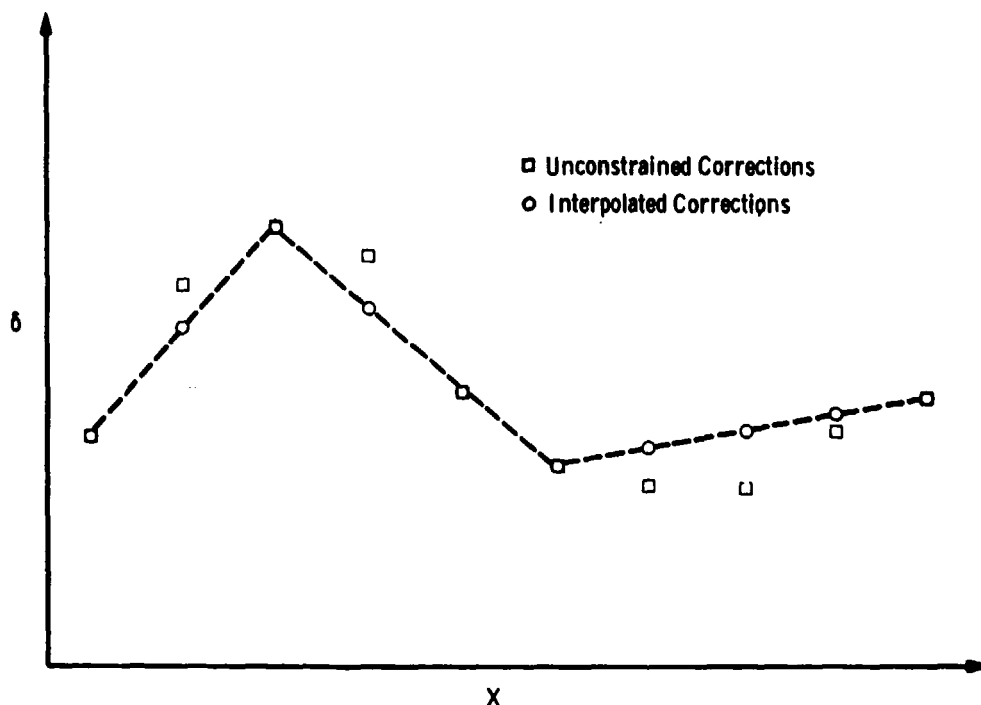


Figure 1. One-dimensional interpolation.

The nearest neighbor principle for 2-D problems is shown in Fig. 2. The coarse grid basis is represented by solid dots. The open circles are interpolated points. The arrows point to the basis components that are used for interpolation. This interpolation can be factored into two 1-D steps. Figures 3a and b show the two steps. The first step reduces from a 5 by 5 grid to a 5 by 3 grid. The second then reduces down to a 3 by 3 grid. The same principle will factor a 3-D problem in three steps.

Interpolation for unequal spacing and irregular geometries is more involved. A convenient alternative is to interpolate as if the geometry were regular with equal spacings. This retains the calculations in a simple form. The result is a "nonlinear form" of interpolation. The longer wavelength information is still passed down to the coarser level. The interpolation errors, as with linear interpolation, are short wavelength in nature and are reduced with the next smoothing pass at the current level.

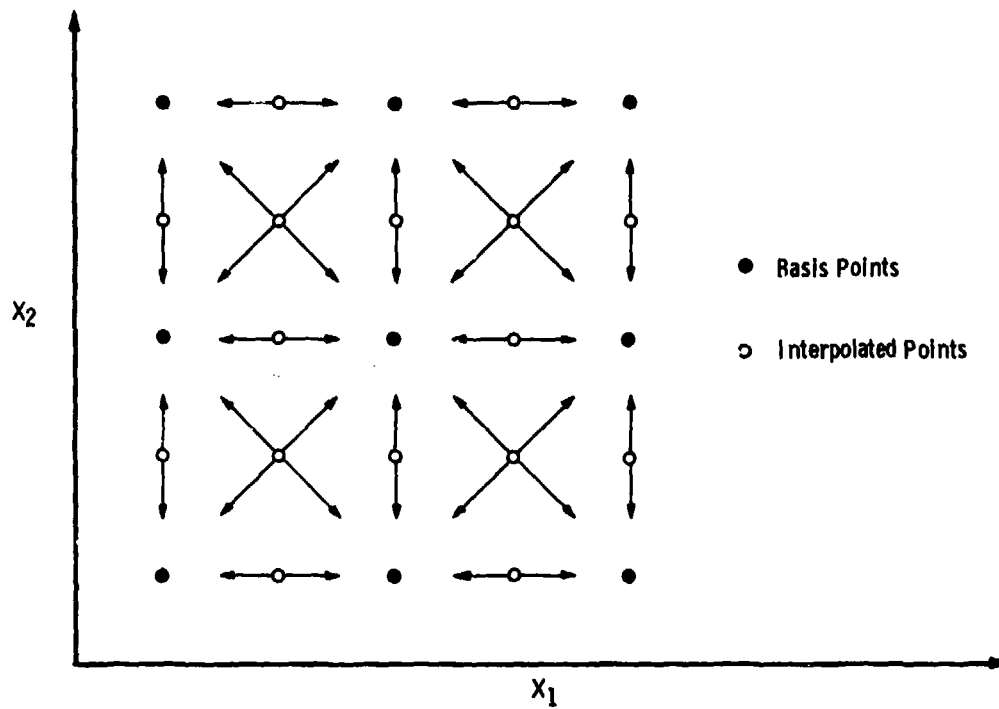


Figure 2. Two-dimensional interpolation.

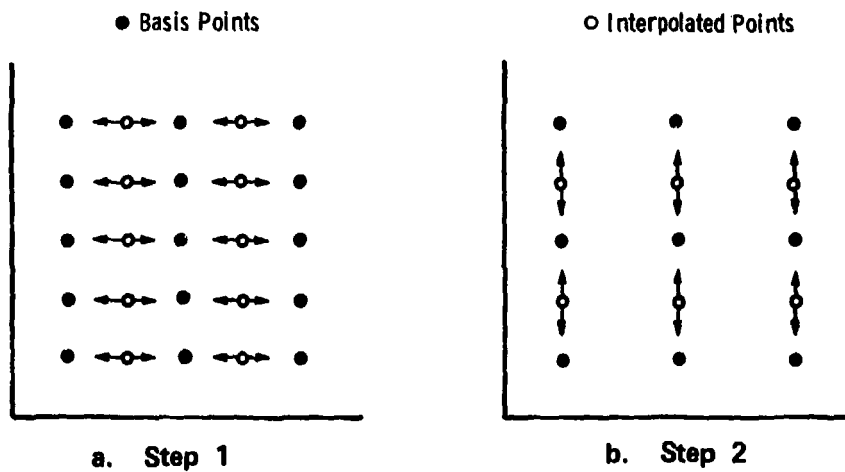


Figure 3. Two-dimensional interpolation in split form.

7.0 SMOOTHING PASS

One of the key elements of the multigrid algorithm is that the wavelength components comparable to grid size must be damped before going to a coarser level. Fortunately, this is the strong point of the conventional iterative methods. The method emphasized in this report is incomplete Crout reduction. Two variations are used in this report. The methods are identical to a complete Crout reduction with the following modifications during the forward pass. In the short version when zeroing an element below the main diagonal, all operations which modify an off-diagonal element are not performed. The result is a quick, efficient iteration which damps out the short wavelength error components. In the long version, all operations which modify the nonzero structured banded elements are kept. All other operations which would modify zero elements are not performed. The long version has better iterative properties at the expense of the additional work required. For the 9-point star 2-D case, the increase in operations is about 60 percent.

8.0 TEST CASE

The preceding methods were used to solve Laplace's equation on a rectangular grid. Dirichlet conditions were imposed at the boundaries. The primary goal of this test case was to verify the method and help in comparing alternatives. The system (2) was obtained using isoparametric quadrilateral finite elements. The basic algorithm consisted of the following. A multigrid cycle of $(m + 1)$ levels was used ($m = 0$ means fine grid only). The coarser levels were obtained by removing every other point in each dimension. Each level contained one smoothing pass (the short version of incomplete Crout followed by a dynamic relaxation). The results (convergence rates) are given in the number of work units required to reduce the error by one order of magnitude. A work unit is defined as the time to set up a fine grid system and make one smoothing pass. It was assumed that the time spent at a lower level was one-fourth that of the next higher level. For comparison with conventional multigrid methods it is also assumed that the time required to compress down to the next lower level is equivalent to that of evaluating the operator at that level. For simple linear problems such as Laplace's or Poisson's equation, the operator evaluation should be quicker. However, for nonlinear problems such as full potential flow, the compression step will probably be faster. The rates given are estimates of the asymptotic rate. They were obtained by iterating until the rates "leveled off." In cases where convergence showed cyclic or erratic behavior, an average of a selected final group of iterations was used. Most of the results are given by 9 by 9, 17 by 17, and 33 by 33 grids with equal spacings. A few results where $n_1 \neq n_2$ and where $\Delta X \neq \Delta Y$ are given at the end of the section.

Table 1 shows the convergence rates for the basic algorithm. Without multigrid levels, the convergence rate rapidly deteriorates with increasing grid size. Using multigrid levels

improves the convergence rates for each grid size, and the results appear independent of grid size. The table indicates that the extra coarse grids are not needed, but nothing is lost by the conservative attitude of using more levels than needed. For comparison purposes the 1.3 convergence rate is equivalent to an error reduction factor of 0.17 per work unit or 0.093 per multigrid cycle. This table should be used as a reference for comparison of alternative methods given in this section.

Table 1 considers the case where the n_1 by n_2 grid points are all interior to the boundary. The boundary conditions had been transferred to the right-hand side of the equations. An alternative would be to include the boundary points as part of the n_1 by n_2 grid. The boundary points have no error, and the neighboring points are interpolated using this zero error boundary. The results are shown in Table 2. The faster convergence at the zero level is due to the fewer non-zero error components. However, where multigrid levels are used, this trend is reversed. The difference is small, though, when sufficient levels are used. Ease of application should probably be the deciding factor.

Table 1. Convergence Rates for Basic Algorithm

n_1 by $n_2 \backslash m$	0	1	2	3	4	5
9 by 9	3.8	1.1	1.2	1.2	---	---
17 by 17	11.5	2.1	1.3	1.3	1.3	---
31 by 31	39.7	6.6	1.8	1.3	1.3	1.3

Table 2. Convergence Rates for Alternate Boundary Conditions

n_1 by $n_2 \backslash m$	0	1	2	3	4	5
9 by 9	2.7	1.3	1.2	1.2	---	---
17 by 17	9.2	2.3	1.2	1.3	1.3	---
31 by 31	34.3	9.5	2.0	1.4	1.4	1.4

Table 3 shows the results when a Jacobi ADI method is used for smoothing. The dynamic relaxer was applied after each of the two ADI sweeps. The multigrid convergence rates are about the same as the incomplete Crout reduction. The main reason for choosing the incomplete Crout was its efficiency. It is easily programmed. For large systems it requires 5 divides (with common divisor), and 12 multiply-add combinations per equation. If applied to a 5-point star system obtained from finite differences, the operation count is three divides and six multiply-adds per equation.

Table 3. Convergence Rates for ADI Smoothing

n_1 by n_2 \ m	0	1	2	3	4	5
9 by 9	6.6	1.4	1.2	1.3	---	---
17 by 17	21.1	3.5	1.5	1.4	1.3	---
31 by 31	76.6	12.0	2.6	1.5	1.4	1.3

Several alternative methods of cycling through the multigrid levels were tried. The order did not significantly change the asymptotic convergence rates. That is, it makes no difference whether one starts with the fine grid and works down to the coarse or vice versa. Attempts at weighting the coarse grid passes were also tried. Table 4 shows results where the number of passes at each level varied linearly from one for level 0 to six for level 5.

Table 4. Convergence Rates for Weighted Passes

n_1 by n_2 \ m	0	1	2	3	4	5
9 by 9	3.8	1.4	1.6	1.6	---	---
17 by 17	11.5	1.4	1.6	1.7	1.7	---
31 by 31	39.7	3.6	1.6	1.7	1.7	1.7

Some improvement is noticed for a few cases where the number of levels is insufficient. However, when sufficient levels of grid are used, equal weighting (one per pass) is better. The degradation of convergence rate is due to the additional work involved. If the rates were given on a "per multigrid cycle" basis, they would be about equal to those using one pass per level. It is important to emphasize that these are asymptotic rates. It was noticed that the weighted cycle was superior in the initial stages. This was attributed to the smooth errors present with the initial guesses. Such a phenomenon is likely with actual problems. Therefore, it is suggested that the early passes be weighted toward the coarser levels with later passes of one per level.

Table 5 shows results without using the dynamic relaxer. While the relaxer made significant improvements at the zero level, only modest gains were achieved when multigrid levels were used. Since its cost is minimal, the dynamic relaxer was retained in the basic algorithm. Its potential gain for other applications may be significant. For example, a 25-percent savings in time was obtained in the multigrid/ADI cases. For comparison purposes, a fixed optimum parameter was determined by trial and error. Convergence rates for the dynamic relaxer and the fixed optimum were essentially the same.

Table 5. Convergence Rates Without Dynamic Relaxer

n_1 by n_2 \ m	0	1	2	3	4	5
9 by 9	6.7	2.0	1.3	1.3	---	---
17 by 17	20.8	5.5	1.6	1.4	1.4	---
31 by 31	73.0	18.6	4.9	1.5	1.4	1.4

Simultaneous relaxation parameters were also tried. The corrections at each level were saved and used as columns of C in Eq. (6). The system of Eqs. (8) can then be solved for the relaxation parameters (components of k). The results were disappointing. Only trivial gains were noticed, not worth the extra work and storage required.

An interesting alternative is to use a constant times the residuals as the smoothers. The dynamic relaxer can be used to determine the unknown constant. Convergence was erratic, with rates in the 2.0 to 4.0 range. For linear problems this rate would be attractive since the time per iteration is minimal. For nonlinear problems, calculating the fine grid system and

compression to coarser levels takes most of the time, and the overall rates would be considerably slower. A problem with this alternative is the relative scaling of each equation.

Several rectangular grids were also tried. The convergence rate for a 9 by 33 grid was in the 1.2 to 1.3 range that was obtained for the square grids. The incomplete Crout smoother is order dependent. That is, different results are obtained depending upon whether the grid is numbered by rows or by columns. The convergence rates for the 9 by 33 grid were essentially unaffected by the direction of node ordering. Ordering along the short dimension gave less than a 2-percent improvement over the other direction.

Figure 4 shows results for varying aspect ratios. A 33 by 33 grid was used in this study. The solid line shows the results using the short, incomplete Crout reductions. The nodes were numbered in the Y direction. As the figure indicates, the convergence rate rapidly becomes impractical for even moderate aspect ratios. Numbering the nodes in the X direction gave the same behavior. The dashed line shows results for the long version of incomplete Crout reduction (nodes numbered in the Y direction). The worst convergence ratio occurs at an aspect ratio of about 10. About twice as many iterations are needed at this aspect ratio. At larger ratios, the convergence rate rapidly improves. Ordering the nodes in

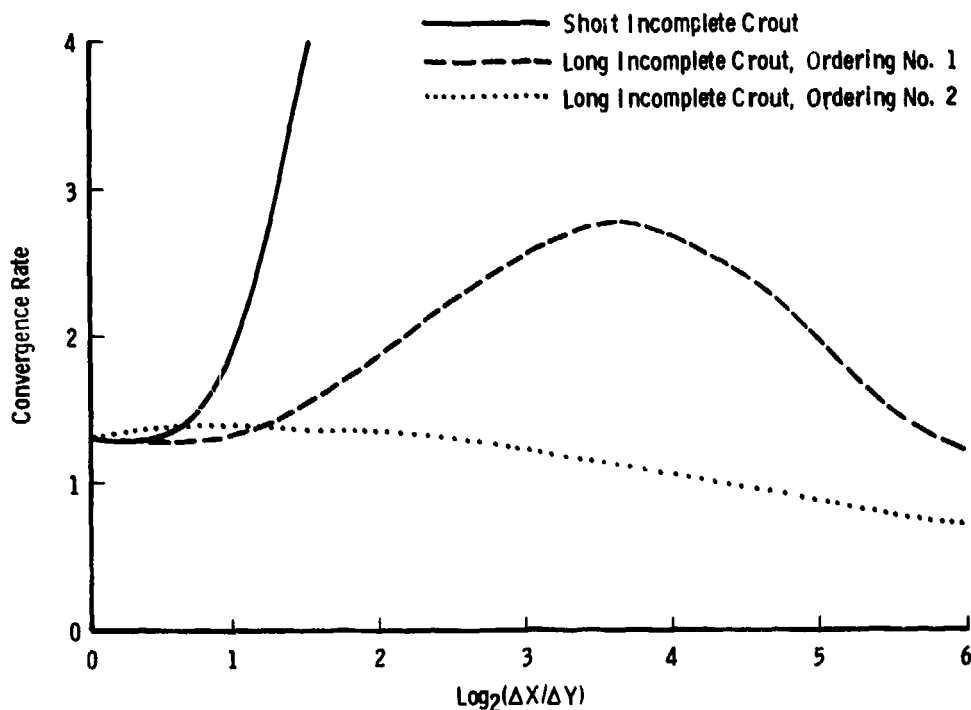


Figure 4. Aspect ratio versus convergence rates.

the X direction gives the results shown by the dotted line. Except for a small increase at the smaller aspect ratios, this ordering gave better results. The unexpected improvement at large aspect ratios is probably due to the regular rectangular geometry. Extrapolation of these results to more general geometries would be speculative. Further study is needed, particularly in the selection of the smoother.

9.0 SUMMARY

A constrained corrections algorithm was described in the previous sections. The method was used to solve Laplace's equation on a rectangle. A convergence rate of 1.3 fine grid work units per decade reduction in error was obtained.

The algorithm uses a multigrid concept with the following components:

1. Incomplete Crout reduction is used to smooth the errors.
2. A dynamic relaxation parameter is used.
3. Coarse grid systems are obtained by constraining the corrections at the fine grid level. These constraints are in the form of simple interpolation.

The method has some drawbacks. The system of equations needs to be stored. Recalculation of the fine grid at each level would increase the computational effort by a factor approximately proportional to the number of levels used. For nonlinear problems, updating the nonlinear parts can be accomplished only at the fine grid level. Another drawback occurs with the simple forms typical of finite difference methods. For example, a 2-D finite difference method usually uses a 5-point rather than a 9-point star. The interpolation used in this paper will not maintain this 5-diagonal system, but expands to a 9-diagonal system.

The main advantage of the method is the influence of the interpolation formulas. The coarse grid systems contain not only the "average residuals," but also fine grid geometry information and the implied interpolation of the solution back to the fine grid. It is expected that this unification between the multigrid phases will prove advantageous when general distorted geometries are used. The method is easy to use and does not require guesswork for determining parameters. Simple interpolation forms are used, producing efficient iterations.

It is the opinion of the author that the advantages will outweigh the disadvantages. A 3-D full potential program is being developed using the method presented in this report.

NOMENCLATURE

A	Coefficient matrix for a linear system of equations
A_i	Coefficient matrix at i th multigrid level
B	Interpolation coefficient matrix
B_i	Interpolation coefficient matrix from level i to level (i - 1)
C	Augmented interpolation coefficient matrix
C_i	Augmented interpolation coefficient matrix from level i to level (i - 1)
C*	Matrix defining linear constraints implied by C
D_i	Interpolation coefficient matrix from level i to fine grid
F	Vector defined by $\partial L / \partial \phi$
k	Vector of unknowns in constrained correction formulation
L	Variational form
L*	Alternate variational form
m	Maximum number of levels used (m = 0 means fine grid only)
n₁, n₂, n₃	Number of nodes in a given direction of the grid
r	Residual vector
r_i	Residual vector at i th level
δ	Correction vector
δ_a	Correction vector approximation
δ_b	A basis vector used for interpolation
δ_i	Basis vector at the i th level
φ	Solution vector
φ_i	i th iteration for the solution vector
()^T	Transpose operator

- $\partial L / \partial \phi$ A vector composed of derivatives of L with respect to elements of ϕ
- $\partial F / \partial \phi$ Matrix composed of derivatives of the elements of F with respect to elements of ϕ (The F components determine rows, and the ϕ components determine columns.)
- $\partial^2 L / \partial \phi^2$ Alternate notation for $\partial^2 F / \partial \phi^2$ (The matrix composed of second derivatives of L with respect to elements of ϕ .)